

## String related functions

```
*&-----
-----*
*& Report  ZDEMO_STRING_EXAMPLE
*&
*&-----
-----*
```

```
REPORT  ZDEMO_STRING_EXAMPLE.
```

```
data : v_name type String, v_name1 type String.
```

```
v_name = ' hi/           dude/           whatz/up'.
write : / v_name color 3.
```

```
TRANSLATE v_name TO UPPER CASE. "/ or LOWER CASE.
```

```
write :/ v_name color 4.
```

```
CONDENSE v_name . " it removes extra gaps
*CONDENSE v_name no-gaps " it removes all gaps .
```

```
v_name1 = v_name.
write : / v_name color 5.
```

```
data v_len type i.
```

```
v_len = strlen( v_name ). " lemnth of string
```

```
write :/ 'length ' color 6, v_len color 6.
```

```
*-----
*-----
```

```
skip 3.
PARAMETER p_sep(1) type c.
```

```
data : v_s1 type String, v_s2 type String, v_s3 type St
```

```

ring, v_s4 TYPE String.
split v_name1 at p_sep into v_s1 v_s2 v_s3 v_s4.

write / 'Split ---->>>>'.
write / v_s1.
write / v_s2.
write / v_s3.
write / v_s4.

data v_name2 type String.
concatenate v_s1 v_s2 v_s3 v_s4 into v_name2 SEPARATED
BY '-'.

write : / ' Concatenate ----->>>>', v_name2 color 7
.

skip 2.

v_name = 'vikasshri77'.
write :/ 'accessing part of string using Offset by 0th
location to 3',v_name+0(3).

write :/ 'accessing part of string using Offset by 2nd
location to 3',v_name+2(3).

*-----
*-----
skip 3.

DATA str1 TYPE String.

str1 = 'hi how are you'.

skip 1.
replace 'you' in str1 with 'you all'.

write :/ 'str replaced ', str1.

skip 1.

str1 = 'hell hell how are you'.

```

```
data v_count type I.
```

```
replace all occurrences of 'Hell' in str1 with 'Hello'  
ignoring case replacement count v_count.  
write :/ 'str replaced all occurrences ', str1 color 5,  
'replacement count -->', v_count color 6.
```

```
skip 1.
```

```
str1 = 'hell hell how are you'.
```

```
replace section offset 7 length 5 of str1 with 'hellooo'  
'.'  
write :/ 'str replaced all occurrences ', str1 color 5.  
skip 3.
```

```
Data : v_str2 type STRING, v_no TYPE i.
```

```
v_str2 = 'hey how are You and how are your parents'.
```

```
FIND ALL OCCURRENCES OF 'how' in v_str2 IGNORING CASE M  
atch COUNT v_no.
```

```
write : / 'how finds ', v_no.  
uline.uline.  
write / v_str2.  
uline.
```

```
skip 3.
```

```
SHIFT v_str2 Left by 3 PLACES.
```

```
write : / 'SHIFT v_str2 Left by 3 PLACES.' COLOR 1.
```

```
write / v_str2.  
ULINE.  
SHIFT v_str2 RIGHT by 2 PLACES.
```

```
write : / 'SHIFT v_str2 Right by 2 PLACES.' COLOR 2.  
write / v_str2.underline.  
SHIFT v_str2 RIGHT CIRCULAR by 3 PLACES.  
write : / 'SHIFT v_str2 RIGHT CIRCULAR by 3 PLACES.' CO  
LOR 3.
```

```
write / v_str2.underline.  
SHIFT v_str2 LEFT CIRCULAR by 2 PLACES.  
write : / 'SHIFT v_str2 Left CIRCULAR by 2 PLACES.' COL  
OR 4.
```

```
write / v_str2.  
underline.underline.
```